Compare

November 30, 2020

1 Compare

This file was used during the train, validate, test loop in order to guage the relative performance of networks. **The main purpose of this file is to study and compare the outputs of the RFDN and RFDN1 models**. In this file, we can zoom in on portions of predicted images to try to understand the differences between upscaled image outputs for the two networks. In this way, we can see how the outputs differ from each other in general, in a qualitative manner.

[1]:

```
# Import modules
from RFDN import RFDN1, RFDN
from compare import Compare, t_test
import numpy as np
```

1.1 Preliminaries

Here, we set up some variables needed for comparison and evaluation purposes.

[2]:

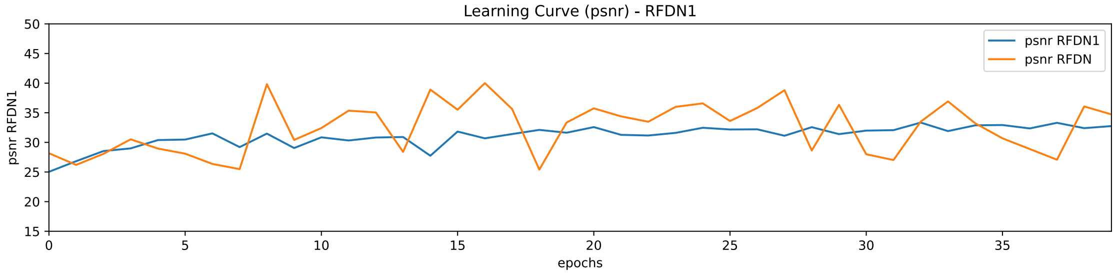
```
model1 = RFDN1(nf=10, upscale=2)
model2 = RFDN(nf=10, upscale=2)
checkpoint_file1 = "/home/samuel/Documents/CMPUT511/Project/Checkpoints/
 ↪AvgLearningCurve/RFDN1/checkpoint_40.tar"
checkpoint_file2 = "/home/samuel/Documents/CMPUT511/Project/Checkpoints/
 ↪AvgLearningCurve/RFDN/checkpoint_2_40.tar"
data_dir = "/home/samuel/Documents/CMPUT511/Project/Data"

comp = Compare(model1, model2, checkpoint_file1, checkpoint_file2, data_dir)
```

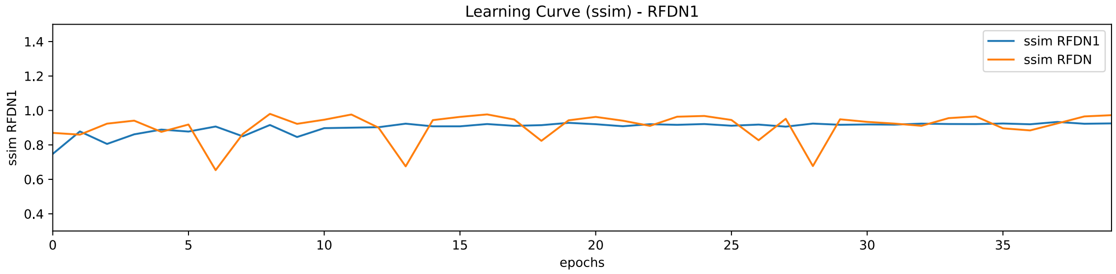
1.2 Learning Curves

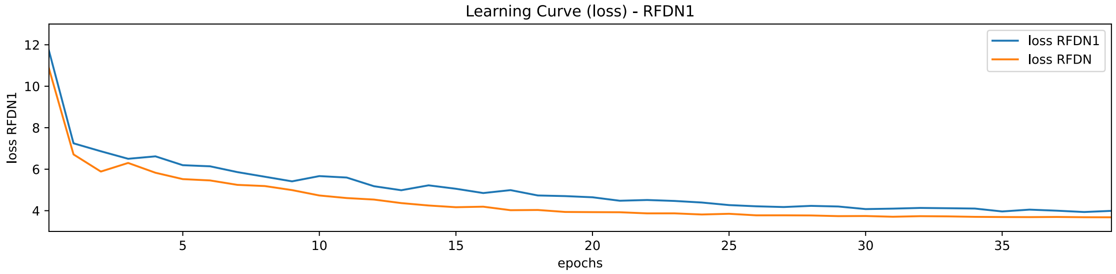
Here, we analyze the learning curves of the two models.

[15]:

```
comp.plot_lc("psnr", x=(0, 39), y=(15, 50), figsize=(15, 3))
comp.plot_lc("ssim", x=(0, 39), y=(0.3, 1.5), figsize=(15, 3))
comp.plot_lc("loss", x=(0, 39), y=(3, 13), figsize=(15, 3))
```







1.3 Save Predictions

Here, we produce image predictions for both networks and print their evaluation metrics. With this line of code, we can upsample and save any upscaled image by both networks.

[17]:

```
image = "/home/samuel/Documents/CMPUT511/Project/Data/val/LR/
 ↪DIV2K_valid_LR_bicubic/X2/0801x2.png"
output_name = "/home/samuel/Documents/CMPUT511/Project/Checkpoints/img_output"
```

[18]:

```
comp.predict(image, img_name=output_name)
```

```
PSNR for model 1 (RFDN1): 35.339031697223646
SSIM for model 1 (RFDN1): 0.9625861644744873
PSNR for model 2 (RFDN): 35.646986149719616
SSIM for model 2 (RFDN): 0.963182270526886
```

```
Average Loss for model 1 (RFDN1):5.093774834459648
Average Loss for model 2 (RFDN): 4.5172790960688145
Saving images
```

2 Average Evaluation Measures

Here, we get the evaluation measures (PSNR, SSIM, inference time) for each validation data instance for each network. We the plot the average evaluation metrics for each type of measure in order to compare the two networks.

[19]:

```
values = comp.get_values()

psnr1 = np.mean(values["psnr"]["model1"])
psnr2 = np.mean(values["psnr"]["model2"])
ssim1 = np.mean(values["ssim"]["model1"])
ssim2 = np.mean(values["ssim"]["model2"])
times1 = np.mean(values["times"]["model1"])
times2 = np.mean(values["times"]["model2"])

print(f"Average inference PSNR for model 1 ({str(model1)}): {psnr1}")
print(f"Average inference PSNR for model 2 ({str(model2)}): {psnr2}\n")
print(f"Average inference SSIM for model 1 ({str(model1)}): {ssim1}")
print(f"Average inference SSIM for model 2 ({str(model2)}): {ssim2}\n")
print(f"Average inference time for model 1 ({str(model1)}): {times1}")
print(f"Average inference time for model 2 ({str(model2)}): {times2}")
```

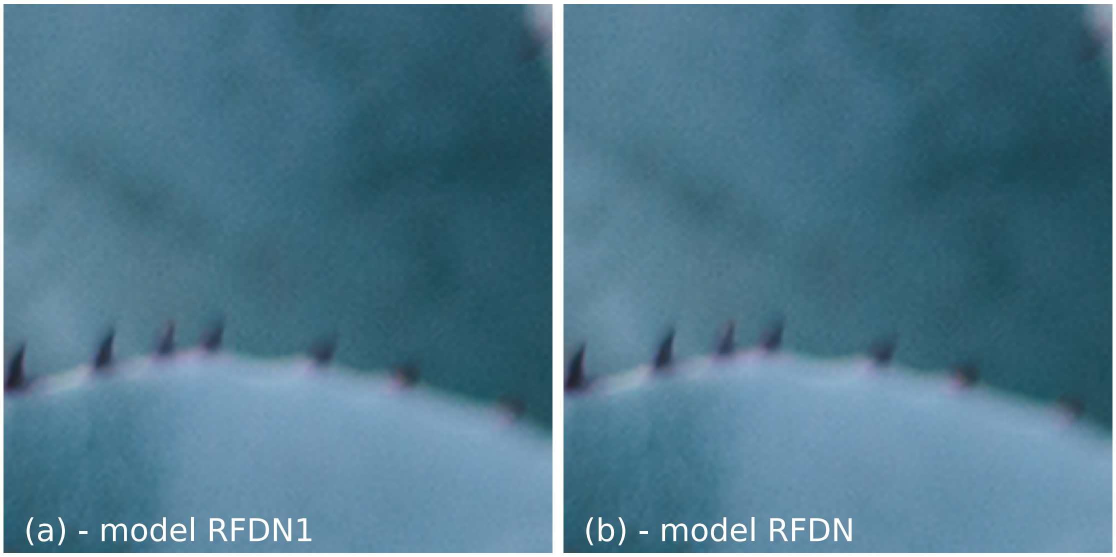
```
100%|      | 100/100 [03:54<00:00,  2.35s/it]Average inference PSNR for
model 1 (RFDN1): 32.93098219938509
Average inference PSNR for model 2 (RFDN): 33.41725744285075

Average inference SSIM for model 1 (RFDN1): 0.9256126284599304
Average inference SSIM for model 2 (RFDN): 0.927844226360321

Average inference time for model 1 (RFDN1): 0.100590660572052
Average inference time for model 2 (RFDN): 0.15677828550338746
```

3 Patch Comparison

In the following cell, we compare patches of image outputs of the two networks we are comparing. In the `compare_patches()` function, the first argument is the image number of the validation data instance, the `size` argument is the size (in pixels) of the resulting patch. The `start` parameter is the coordinates of the starting pixel. For more documentation, see the .py file `compare.py`.

[6]:

```
comp.compare_patches(56, size=250, start=(110,200))
```



4 Statistical significance test

Here, we test to see if the PSNR, SSIM, and Loss values are statistically significant from each other. If they are, then we can say that most likely, one of the two models are better than the other. If they are not statistically significant, then we cannot say if any model is better.

```
[7]: checkpoint_dir1 = "/home/samuel/Documents/CMPUT511/Project/Checkpoints/
      ↪AvgLearningCurve/RFDN1"
     checkpoint_dir2 = "/home/samuel/Documents/CMPUT511/Project/Checkpoints/
      ↪AvgLearningCurve/RFDN"

     p_psnr = t_test(checkpoint_dir1, checkpoint_dir2, "psnr")
     p_ssim = t_test(checkpoint_dir1, checkpoint_dir2, "ssim")
     p_loss = t_test(checkpoint_dir1, checkpoint_dir2, "loss")

     print(f"P-value for PSNR: {p_psnr}")
     print(f"P-value for SSIM: {p_ssim}")
     print(f"P-value for Loss: {p_loss}")
```

```
P-value for PSNR: 0.15422987536059868
P-value for SSIM: 0.24082446630607615
P-value for Loss: 0.05905154873981436
```